

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**APPARATUS AND METHOD FOR
AUTOMATICALLY POSITIONING
A CURSOR ON A CONTROL**

Inventor(s):
Michael B. Robin

ATTORNEY'S DOCKET NO. MS1-206USC1

030522589

1 **RELATED APPLICATIONS**

2 This is a divisional of U.S. Patent Application Serial No. 08/209,166, filed
3 March 10, 1994, which is now U.S. Patent No. 5,808,604.

4 **TECHNICAL FIELD**

5 The present invention relates to the field of computer-controlled displays
6 and, more particularly, to an apparatus and method for designating control
7 groupings and automatically positioning a cursor on a control.

8
9 **BACKGROUND OF THE INVENTION**

10 Computer-controlled displays are currently in widespread use in both home
11 and business environments. Moreover, due to the price drop of microprocessors in
12 recent years, computer-controlled displays are being used in environments other
13 than conventional personal computers. For example, computer-controlled displays
14 are often used in home entertainment systems (e.g., VCRs, video games or other
15 types of interactive television systems). However, because these systems typically
16 lack both sophisticated pointing devices and full-size keyboards, cursor control is
17 problematic. Cursor control refers to the ability of a user to move a cursor to a
18 specified location on the computer-controlled display. The term "cursor" as used
19 herein includes any form of movable marker such as an arrow, a screen highlight,
20 a screen focus, or the like.

21 Figure 1 illustrates a computer system 105 of a typical home entertainment
22 device. Typically, the home entertainment device features a computer 110, an
23 input device 120, and a video display device 170. Additionally, the computer 110
24 contains a receiver 130, a memory 140, a central processing unit (CPU) 150, and a
25 display driver 160. These components are interconnected as shown.

1 The input device 120 includes a directional control device 122 (e.g., a
2 joystick, an arrow keypad, a mouse) and a control switch 124 (e.g., an action
3 button, a function button, or a mouse click-button). By actuating the directional
4 control device 122, a user can move a cursor 172 on the screen of the video
5 display device 170. Similarly, by releasing the directional control device 122, the
6 user terminates cursor movement. The phrase "releasing the directional control
7 device" as used herein refers to any process for deactivating a directional control
8 device 122, so that the directional control device 122 no longer effects cursor
9 movement. The screen of the video display device is logically divided into
10 discrete points. Each discrete point is addressable by Cartesian coordinates (X,
11 Y). The screen of the video display device 170 displays a number of controls 174.
12 Each control 174 is a child window that occupies a defined area of the video
13 display device 170 and is used to input instructions to a computer program 142
14 running on the computer system 110. Typically, the computer program 142 is
15 contained within the memory 140 of the computer system 110. Generally, the
16 controls are maintained in one of two states, (1) active (indicating that an option
17 represented by the control 174 is currently available to the user), or (2) inactive
18 (indicating that an option represented by the control 174 is currently unavailable to
19 the user).

20 Typically, a software programmer decides which controls of a given
21 window should be active or inactive, depending upon the state of the program.
22 For example, in the VCR Programming Options Window 205 of Figure 2A, the
23 controls 220, 222 and 224 with double borders are active while the controls 230,
24 232 and 234 with the single borders are inactive. In this case, the Play
25 control 230, Fast Forward control 232 and Rewind control 234 are inactive due to

1 the "ON" status of the Timer ON/OFF control 220. In conventional systems, the
2 software programmer has to individually activate or deactivate a given control. To
3 disable only a fraction of the currently active controls, the programmer selectively
4 deactivates each of the controls. Such selective deactivation is cumbersome.

5 To invoke an option represented by an activated control, the user positions
6 the cursor 172 on the desired activated control and actuates the control switch 124.
7 By actuating the control switch 124 (*e.g.*, by depressing an action button), the user
8 directs the computer program 142 to execute the desired option. However, as
9 previously discussed, most home entertainment systems do not have a
10 sophisticated pointing device (such as a mouse) for a directional control device
11 122. Instead, the directional control device 122 is frequently a joy stick or a
12 simple four-directional control device. As a result, positioning the cursor 172 on
13 the desired control 172 is a difficult process for the user.

14 In conventional systems, positioning the cursor is accomplished by
15 "tabbing" or "free-roaming." When tabbing is used, the computer system 110
16 maintains a hard-wired list of the controls 174. The hard-wired list of controls
17 designates the sequence in which the user must traverse through the activated
18 controls. As previously explained, Figure 2A illustrates the Timer ON/OFF
19 control 220, the Timer Set control 222, and the Set Clock control 224 as active,
20 while illustrating the Play control 230, the Fast Forward control 232 and the
21 Rewind control 234 as inactive. Thus, in order for a user to access the Set Clock
22 control 224, the user must tab through the Timer ON/OFF control 220, and Timer
23 Set control 222 before reaching the Set Clock control 224. As a result, the
24 conventional process of tabbing is inflexible and is highly frustrating if the user
25 accidentally tabs over a desired control.

1 "Free-roaming" is also a frustrating process when a crude positioning
2 device is used. In free roaming, the user moves the cursor towards a desired
3 control using the directional control device 122. However, because the directional
4 control device 122 is typically a crude positioning device, the actual process of
5 positioning the cursor on a control is difficult. For example, it is not uncommon
6 for the user to over-shoot or under-shoot the desired control before finally placing
7 the cursor on the desired location. Thus, neither tabbing nor free-roaming provide
8 a desirable technique to position a cursor on a desired control of a computer-
9 controlled display.

10 Positioning a cursor in conventional systems has a further complication.
11 This complication arises because active controls frequently become obscured by
12 subsequently displayed items. For example, the Set Time Window 250 of Figure
13 2B obscures all the active controls 260 (shown in phantom form) of the VCR
14 Programming Options Window 205. Under these conditions, conventional
15 systems allow the user to activate the obscured controls 260 as well as the
16 unobscured controls 251, 252, 253 and 254. Such an occurrence is undesirable as
17 the user can inadvertently activate one of the obscured controls.

18 19 **SUMMARY OF THE INVENTION**

20 The preferred embodiment of the present invention is a control manager for
21 activating and deactivating controls as well as positioning a cursor on an ideal
22 control. In its preferred embodiment, the present invention allows a programmer
23 to quickly and easily activate and deactivate controls by designating a plurality of
24 controls as a "control grouping." Subsequently, the preferred embodiment of the
25 present invention designates an identifier for the entire control grouping. The

1 preferred embodiment of the present invention then quickly activates and
2 deactivates the entire grouping by toggling a state of the identifier.

3 In the preferred embodiment of the present invention, the user navigates
4 around a screen of a computer-controlled video display device by actuating a
5 directional control device. Upon releasing the directional control device, the
6 preferred embodiment of the present invention automatically positions the cursor
7 on an ideal control. The preferred embodiment determines the ideal control by
8 examining the state of an application program and the last actions of the user.
9 More specifically, for each active control, the preferred embodiment determines if
10 the control is (1) obscured by a subsequently displayed item or (2) positioned at a
11 location that the application program has specified as invalid.

12 The preferred embodiment determines if the control is obscured by a
13 subsequent item of data by comparing a registrant identifier with a current
14 identifier. The registrant identifier identifies a portion of an application program
15 that registered the control with the control manager. The current identifier
16 identifies the entity (*i.e.*, the portion of the application that registered the control)
17 that currently surrounds the Cartesian coordinates that define the control. For
18 brevity, the Cartesian coordinates that define the control will be referred to as the
19 "control position." If the registrant identifier does not match the current identifier,
20 the preferred embodiment recognizes that the control is obscured. The preferred
21 embodiment implicitly designates all obscured controls as controls which do not
22 constitute a valid resting position for the cursor.

23 The preferred embodiment determines if a control is positioned at a
24 location that the application program has specified as invalid by comparing a last
25 cursor position with the control position. The last cursor position is defined by the

Cartesian coordinates that the cursor pointed to when the user released the directional control device. The preferred embodiment then calculates a Control Distance and a Control Angle. The Control Distance is the absolute distance of a control path: the control path being a path which connects the last cursor position with the control position. The Control Angle is the angle formed between the control path and a Last Direction of Cursor Movement Path. The preferred embodiment compares the calculated Control Distance and Control Angle Values with respective Skip Distance and Skip Angle Values. These skip distance and skip angle values are specified by the application program. Subsequently, the preferred embodiment does not consider any control which has either a calculated Control Distance or Control Angle which is greater than the respective Skip Distance or Skip Angle.

At this point, the preferred embodiment calculates a weighted distance for each remaining control. Each weighted distance is calculated as a function of the Control Distance, the Control Angle, and an Angle Importance Weight. More specifically, the Weighted Distance is directly proportional to both the Control Distance and the Control Angle. Additionally, as the Angle Importance Weight increases, the effect given to the Control Angle in the Weighted Distance calculation is diminished while the effect given to the Control Distance is increased. After completing a Weighted Distance calculation for each remaining control, the preferred embodiment of the present invention positions the cursor on the control having the smallest Weighted Distance. In this manner, the preferred embodiment of the present invention automatically positions the cursor on the ideal activated control.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a conventional computer system.

Figure 2A illustrates a plurality of controls that are displayed on the video display device of Figure 1.

Figure 2B illustrates the plurality of controls of Figure 2A after they have been obscured by a subsequently displayed item of data.

Figure 3 is a block diagram of a computer system for practicing the preferred embodiment of the present invention.

Figure 4 illustrates a control position and status data structure for use in the preferred embodiment of the present invention.

Figure 5 is a functional block diagram illustrating the main functional components of the preferred embodiment of the present invention.

Figure 6 is a flow diagram illustrating an overview of the steps performed in the preferred embodiment of the present invention.

Figure 7A illustrates the steps performed in a process of activating and deactivating control groupings in the preferred embodiment of the present invention.

Figure 7B illustrates the steps performed in a process of positioning a cursor in the preferred embodiment of the present invention.

Figure 8 is a flow diagram illustrating the steps performed in the steps performed in a process for registering controls in the preferred embodiment of the present invention.

Figure 9 is a flow diagram illustrating the steps performed in a process for activating and inactivating hot spots in the preferred embodiment of the present invention.

1 Figure 10 is a flow diagram illustrating the steps performed in a process for
2 positioning a cursor in the preferred embodiment of the present invention.

3 Figure 11 is a flow diagram illustrating the steps performed in a process for
4 determining the ideal control upon which to position the cursor.

5 **DETAILED DESCRIPTION**

6 The preferred embodiment allows the user to initiate cursor movement by
7 actuating a directional control device. When the user releases the directional
8 control device, the preferred embodiment determines the position on the screen
9 that the cursor is pointing to. Subsequently, the preferred embodiment
10 automatically moves the cursor from the determined position to an ideal control.
11 In this manner, the preferred embodiment allows the user to quickly jump to a
12 desired control without forcing the user to actually navigate the cursor over the
13 desired control cursor in this fashion. Such positioning is particularly
14 advantageous when the user has access only to a crude positioning device (*e.g.*, a
15 joystick or an arrow keypad). Additionally, by permitting a programmer to
16 designate a plurality of controls as a single control grouping, the preferred
17 embodiment allows an application program to quickly activate and deactivate a
18 selected number of the controls. Such selective activation and deactivation saves
19 the application programmer from making repetitive calls to a control manager.
20 Furthermore, the preferred embodiment provides a convenient method and
21 apparatus for prohibiting access to controls that are covered by subsequently
22 displayed windows or other items of data. More specifically, for each point which
23 defines a registered control, the preferred embodiment compares a registrant
24 identifier with a current identifier. In this manner, the preferred embodiment
25 determines if a control has been obscured by a subsequently displayed item of

1 in other areas of the computer system (*e.g.*, in the device driver 360) without
2 departing from the spirit and scope of the present invention.

3 Figure 4 illustrates a data structure 400, associated with an application
4 program. As will be explained below, the preferred embodiment of the present
5 invention maintains this data structure to determine the ideal control of the
6 application program. Those skilled in the art will appreciate that the data structure
7 of Figure 4 represents only one of many possible methods for storing the data
8 explained below. For example, instead of storing the data in an array format as
9 illustrated in Figure 4, the data may be stored in a linked-list structure or stored in
10 separate fields which are not aggregated together. For convenience, the data
11 structure 400 of Figure 4 illustrates the data used by the preferred embodiment of
12 the present invention in a single cohesive grouping. This format was chosen to
13 enhance the readability of the detailed description. As such, the format in which
14 the data is presented in Figure 4 should not be used to limit the scope of the
15 invention.

16 The data structure 400 is used for managing the activation of controls 374
17 and positioning the cursor 372 on a control 374. The data structure includes a
18 Control Word 405 for quickly activating and deactivating groupings of controls.
19 The Control Word 405 includes a number of Control Grouping Indicators 406.
20 Each Control Grouping Indicator 406 defines the status (active or inactive) for a
21 number of aggregated controls. This number of aggregated controls is referred to
22 as a control group. Additionally, the data structure includes three parameters
23 which are specified by an application program 346: a Skip Angle Parameter 410,
24 a Skip Distance Parameter 420, and an Angle Importance Weight Parameter 430.
25 As will be explained below, the control manager 345 utilizes these three

1 parameters to determine an ideal control 374 upon which to position the cursor
2 372. The data structure also includes a Control Position and Status Indicator 440
3 (CPSI) for each control. Each CPSI contains: (1) a Position Identifier 441 for
4 identifying a position of a control on the screen (*e.g.*, a set of Cartesian
5 coordinates representing the control); (2) a Registrant Identifier 442 for identifying
6 the entity which registered the control (*e.g.*, a window handle identifying a
7 registering window); (3) an Active Flag 443 for indicating an active/inactive state
8 of the control; and (4) Control Group Identifiers 444 for identifying a membership
9 of the control in one or more control groups.

10 As will be explained below, the control manager 345 inputs, from
11 application programs 346 and 348, values for the Control Word 405, the Skip
12 Angle Parameter 410, the Skip Distance Parameter 420, the Angle Importance
13 Weight Parameter 430, the Position Identifier 441, and the Registrant Identifier
14 442. The preferred embodiment of the present invention then uses these inputted
15 values to dynamically maintain the remaining contents of each CPSI (*e.g.*, the
16 Active Flag 443 and the Control Group Identifiers 444).

17 Figure 5 is a functional block diagram illustrating the main software
18 processes of the preferred embodiment of the present invention. The preferred
19 embodiment of the present invention registers controls (block 501) by
20 (1) inputting positioning information, a registrant identifier, and control group
21 numbers from an application program 346 (block 502) and (2) storing the inputted
22 data in the respective fields of a CPSI 440 (block 503). Similarly, the preferred
23 embodiment activates and deactivates controls (block 510) by (1) inputting control
24 information (block 511) and updating the Active Flags 443 of the CPSI
25 accordingly (block 512). More specifically, the preferred embodiment may either

1 (1) activate/deactivate individual controls by inputting individual control
2 identifiers and active/inactive status information (block 513), or
3 (2) activate/deactivate an entire control grouping by inputting a mask and
4 superimposing the mask into the Control Word 405 (block 514). As will be
5 explained below in reference to Figure 9, this mask is a grouping of binary
6 numbers which are used to selectively set or reset the Control Grouping Indicators
7 406.

8 The preferred embodiment of the present invention positions the cursor on
9 an ideal control (~~block 515~~^{blocks 515 & 516}) by removing from consideration any controls that are
10 (1) obscured by subsequently displayed items of data (block 517), or
11 (2) positioned at distances or angles which, relative to the cursor's position and
12 path, are greater than respective application-defined maximum values (blocks 518
13 and 519). The preferred embodiment calculates a Weighted Distance for each
14 control which is not removed from consideration^(block 520). The preferred embodiment
15 calculates the Weighted Distance by first identifying a Last Direction of Cursor
16 Movement Path and a Last Cursor Position. The Last Direction of Cursor
17 Movement Path defines the direction that the user was moving the cursor 372
18 immediately before the user released the directional control 322. Similarly, the
19 Last Cursor Position defines the Cartesian coordinates of a cursor pointer when the
20 user released the directional control 322.

21 Having identified the Last Direction of Cursor Movement Path and the Last
22 Cursor Position, the preferred embodiment then calculates the Weighted Distance
23 for the control as a function of that control's Control Distance and Control Angle.
24 As stated above, a control's Control Distance is the distance that connects the Last
25 Cursor Position with the Control Position. The line which defines the Control

Distance is referred to as the Control Path. Since both the Last Cursor Position and the Control Position are typically related by Cartesian coordinates, the Control Distance can be calculated in a straightforward manner (*e.g.*, via a Euclidean or manhattan distance calculation). A control's Control Angle is the angle formed between the Last Direction of Cursor Movement Path and the Control Path. Generally, the Weighted Distance of a control is directly proportional to both the Control Angle and the Control Distance. Additionally, the preferred embodiment allows the user to selectively minimize or maximize the effect given to either the Control Angle or the Control Distance by varying the previously described Angle Importance Weight 430. As will be appreciated by those skilled in the art, a variety of formulae for calculating the weighted distance of a control may be implemented without departing from the spirit or scope of the present invention. One example is the following formula which is used in the preferred embodiment of the present invention:

$$\text{WeightedDistance} = \text{ControlDistance} + \frac{\text{ControlAngle} * \text{Control Distance}}{\text{AngleImportance Value}}$$

After calculating Weighted Distances for each control, the preferred embodiment of the present invention positions the cursor on the control with the smallest associated Weighted Distance (block 521).

Having described an overview of the preferred embodiment of the present invention, the respective components of the preferred embodiment will now be explained in detail. The detailed discussion to follow illustrates the preferred embodiment as it is incorporated in an interactive television system. In this

1 example, the input device 320 of Figure 3 is a remote control and the directional
2 control device 322 is assumed to be a four-directional arrow keypad.
3 Additionally, in this example, the computer 310 is a VCR and the video display
4 device 370 is a television. However, the present invention is not limited to any
5 particular directional control device 322, computer system 310, or video display
6 device 370.

7 Figure 6 shows an overview of the steps performed by the control manager.
8 In order for the preferred embodiment to direct processing control along a "YES"
9 path of a decision step (*e.g.*, step 610, 630, 660, or 680), the preferred embodiment
10 must determine that the application program has requested the corresponding
11 action. Typically, the application program 346 notifies the Control Manager
12 Program that a given action is desired. As will be appreciated by those skilled in
13 the art, such notification may be accomplished in any one of a number of ways
14 (*e.g.*, setting a request flag, calling components of Figure 6 directly, or a hardware
15 interrupt). The actual means of notification does not comprise a limit on the
16 present invention.

17 The preferred embodiment begins processing by determining if the
18 application program requests to register controls (step 610). The example assumes
19 that a VCR user has just activated a VCR Options Program 346 by depressing a
20 VCR programming option switch on the VCR remote control device. As such, the
21 VCR Options Program 346 generates a VCR Programming Options Window 705
22 of Figure 7A on the video display device 370. Figure 7A illustrates controls in an
23 identical manner to that previously illustrated in regards to conventional systems
24 (Figure 2A). However, as will be explained below the preferred embodiment of
25 the present invention discloses a novel technique for activating and deactivating

the controls of Figure 7A. After generating the window, the VCR Options Program 346 registers each control (720, 722, 724, 730, 732 and 734) in the VCR programming window 705 with the control manager 345 (step 620).

The Register Control Program of Figure 8 accepts controls from an application program 346 and registers them in the data structure 400. The Register Control Program begins by determining if the application program has an additional control to register (step 810). If the application program has an additional control to register, the preferred embodiment inputs the respective control data (step 820). In the example, the preferred embodiment of the present invention inputs control data for the controls (720, 722, 724, 730, 732 and 734) of the VCR Options Program 346 (step 820). The preferred embodiment then registers the controls by storing the inputted control data in the previously described CPSIs 440 (step 830). More specifically, for each control, the preferred embodiment stores a set of Cartesian coordinates which identifies the center of the control in a respective Position Identifier 441. Additionally, the preferred embodiment stores a window handle for each control in the respective Registrant Identifiers 442. A window handle is a numerical identifier that identifies the registering window (*i.e.*, the VCR Programming Options Window 705). By default, the preferred embodiment also sets the Active Flag 443 for each newly registered control. However, the application program 346 can request that this flag be initially reset. Furthermore, the preferred embodiment stores control group numbers for each control in respective Control Group Identifiers 444. These group numbers indicate the control groupings to which the controls belong. As will be explained below, the application may designate a control to be considered as an element of a control grouping at any time. For purposes of

example, assume that the VCR Options Program 346 specifies that the Timer ON/OFF control 720, the Timer Set control 722, and the Clock Set control 724, all belong to a first grouping of controls and the Play control 730, the Fast Forward control 732, and the Rewind control 734, all belong to a second grouping of controls. As a result, the Register Controls Program updates the CPSIs 440 to reflect the membership of the controls in their respective groups. In the example, the preferred embodiment of the present invention stores either a one or a two in the Control Group Identifier 444 of each control depending on whether the associated control belongs to the respective first or second grouping. After inputting and storing this data for each control, the Register Control Program of Figure 8 returns to the Control Manager Program of Figure 6.

The Control Manager Program of Figure 6 continues processing by determining if the VCR Options Program 346 has requested to set or modify the parameters used in the Weighted Distance calculation (*i.e.*, the Skip Angle Value 410, the Skip Distance Value 420, and the Angle Importance Weight 430) (step 630). By varying values for these parameters, an application programmer informs the control manager 345 how to determine an ideal resting position for the cursor 372. For example, since the preferred embodiment uses

$$\text{Weighted Distance} = \text{Control Distance} + \frac{\text{Control Angle} * \text{Control Distance}}{\text{Angle Importance Value}}$$

to calculate weighted distances, an application programmer can direct the control manager 345 to give predominate weight to controls which lie in proximity to a Last Direction of Cursor Movement Path (as opposed to controls which are angled away from the direction of Last Direction of Cursor Movement Path) by selecting

1 a large Skip Distance Value 420, a small Skip Angle Value 410, and a large Angle
2 Importance Weight Value 430.

3 Typically, the application programmer heuristically determines values for
4 the above parameters based on the size of the window, the number of controls in
5 the window and the relative locations of the controls to each other. In this case,
6 assume that the application programmer has determined that a Skip Angle Value
7 of 175°, a Skip Distance Value of 450 units, and an Angle Importance Weight
8 Value of 100 represent ideal values for the VCR Programming Options Window
9 705. The effect of the selection of these values will be discussed in detail below
10 relative to Figures 10 and 11. Under these conditions, the preferred embodiment
11 receives the values from the VCR Options Program 346 (steps 630 and 640) and
12 stores the inputted values in their respective locations of the data structure 400
13 (step 650).

14 At this point, the Control Manager Program of Figure 6 determines if it
15 should activate or deactivate any controls (step 660). For purposes of example,
16 assume that the VCR Options Program 346 originally activated all controls of the
17 VCR Programming Options Window 205. Further assume that the user
18 subsequently set a timer of the VCR which renders normal play options (*i.e.*, play,
19 fast forward and rewind) unavailable. In this case, the unavailable options are
20 represented by the controls of the second control grouping (730, 732, and 734).
21 Under these conditions, the VCR Options Program requests deactivation of the
22 second control grouping. As a result, the Control Manager Program transfers
23 control to the Activate/Deactivate Control Program of Figure 9 (step 670).

24 The Activate/Deactivate Control Program of Figure 9 activates or
25 deactivates controls as requested by the application program. The

1 Activate/Deactivate Control Program expects a mask as input if the application
2 program 346 is requesting to activate or deactivate a grouping of controls. The
3 mask is a grouping of binary numbers which are used to selectively screen out or
4 let through certain bits in the Control Word 405. As will be explained below, the
5 preferred embodiment performs a logical AND operation using the inputted mask
6 and a current value of the Control Word 405 to selectively set or reset the Control
7 Grouping Indicators 406 of the Control Word 405. Alternatively, if the
8 application program is requesting to activate or deactivate an individual control,
9 the Activate/Deactivate Control Program expects (1) control identifier (*e.g.*, a
10 CPSI array element number) and (2) an activate/deactivate flag, as input.

11 In the example, the preferred embodiment of the present invention
12 determines that the VCR Options Program 346 is requesting to activate or
13 deactivate the second control grouping (step 910). Alternatively, had the
14 application requested to activate or deactivate an individual control, the preferred
15 embodiment of the present invention (1) receives the individual control identifier
16 and the activate/deactivate flag (step 920) and (2) appropriately activates or
17 deactivates the identified control (step 930). Back to the example, the preferred
18 embodiment inputs a mask to deactivate the second control grouping (step 940).
19 For purposes of illustration, assume that the second control grouping is
20 represented by the second bit 406 of a Control Word 405 and that a null value for
21 the bit deactivates a grouping: thus, the inputted mask would appear as represented
22 by 945 in Figure 9 (*i.e.*, the last four bits of the inputted mask are 1101 or 0xD in
23 hexadecimal notation). Subsequently, the preferred embodiment of the present
24 invention superimposes the inputted mask 945 into Control Word 405 using a
25 logical AND operation (step 950) of Figure 9. Having masked the Control Word

1 405 of Figure 4, the preferred embodiment updates the Active Flag 443 of each
2 CPSI 440 in accordance with the newly masked bit values of the Control Word
3 405. In this case, the preferred embodiment resets each Active Flag 443 for each
4 CPSI having a control group number of two stored as a Control Group Identifier.
5 Having updated each relevant Active Flag 443, the Activate/Deactivate Controls
6 Program of Figure 9 returns control the Control Manager Program 6. By
7 representing a grouping of controls as a single bit of the Control Word 405, the
8 preferred embodiment of the present invention provides a convenient method for
9 an application program to activate or deactivate a fraction of its registered
10 controls. By using a mask and a control word, the preferred embodiment allows
11 the application programmer to activate/deactivate the desired fraction controls
12 with a minimum amount of coded instructions (*i.e.*, by merely loading a mask
13 value into a parameter as opposed to individually specifying each control that the
14 application programmer wants activated or deactivated). At this point, the
15 example has demonstrated how the preferred embodiment of the present invention
16 registers controls and quickly deactivates a fraction of formerly activated controls.
17 The example will now demonstrate the innovative automatic positioning technique
18 of the preferred embodiment of the present invention.

19 Figure 7B represents a display of the video display device 370 of Figure 3.
20 More specifically, Figure 7B illustrates the VCR Programming Window 705 of
21 Figure 7A after it has been obscured by the Subsequently Displayed Window 750
22 of Figure 7B. The active controls of Figure 7A are represented by the controls
23 displayed in phantom form 760 in Figure 7B. Additionally, assume that the
24 Subsequently Displayed Window 750 contains four active registered controls 751,
25 752, 753, and 754 (registered and activated in the previously described manner,

Figures 8 and 10). For purposes of example, assume that the user was moving the cursor in a last direction of movement represented by a Last Direction of Movement Path 770. Additionally, the position on the screen of the video display device 370 at which the user terminated the movement of the cursor is represented by 780 in Figure 7B. The Control Manager Program of Figure 6 determines that the user terminated cursor movement by releasing the directional control 322 (step 680) and transfers control to the Position Cursor Program of Figure 11 (step 690).

The Position Cursor Program of Figure 10 determines the ideal control to position the cursor on after the user has released a switch of the directional control device 322 (*e.g.*, if the user has released a previously depressed directional arrow button). In the example, the Position Cursor Program begins by determining the Last Cursor Position (*e.g.*, the Cartesian coordinates for the position on the screen pointed to by the cursor when the user released the directional control 322) (step 1010). Additionally, the Position Cursor Program determines a Last Direction of Movement Path for the cursor (step 1020). Typically, both the Last Cursor Position and the Last Direction of Movement Path are readily available as parameters contained within the display driver 360. Having determined the point of release and the Last Direction of Movement Path, the Position Cursor Program calls a Determine Ideal Control Program of Figure 11 (step 1030).

The Determine Ideal Control Program of Figure 11 determines the optimal control upon which to position the cursor. The preferred embodiment begins the determination process by setting a Current Minimum Weighted Distance Parameter to an arbitrarily large value (step 1105). For purposes of example, assume that the Current Minimum Weighted Distance Parameter is initially set to 65,535. Subsequently, the preferred embodiment examines the positioning and

status of each control (steps 1110-1170). In the example, there are seven active controls: the unobscured controls of the Subsequently Displayed Window 750 and the obscured controls 760 of the VCR Programming Options Window 705. In the interests of brevity, the three obscured controls 760 of window 705 will be treated in a single pass of the loop formed by steps 1110 and 1120 of Figure 11. The preferred embodiment of the present invention determines that controls 760 are obscured by retrieving the Cartesian coordinates stored in the Position Identifier 441 of the CPSI. Having retrieved the Cartesian coordinates, the preferred embodiment of the present invention passes the retrieved coordinates to a Get Window From Point Program. In turn, the Get Window From Point Program returns the active window handle that surrounds the point represented by the passed center coordinate. In the example, the Get Window From Point Program returns the window handle of the Subsequently Displayed Window 750. The preferred embodiment then compares the subsequently displayed window handle with the window handle stored as the Registrant Identifier 442 for the associated control. As previously mentioned, a window handle is a unique numerical identifier of a window. As such, the window handle of the VCR Programming Options Window 705 does not match the window handle of the Subsequently Displayed Window 750. Consequently, the preferred embodiment does not consider any of the obscured controls 760 as valid resting positions for the cursor (steps 1110 and 1220).

Subsequently, the Determine Ideal Control Program examines the AM/PM control 754 of Figure 7B (steps 1110-1130 of Figure 11). In this case, however, the angle formed between (1) the Last Direction of Cursor Movement Path 770 and (2) the AM/PM Control Path 774, is equal to 180°. As previously discussed,

1 the application programmer specified the skip angle to be 175°. Therefore, the
2 preferred embodiment does not consider the AM/PM control 754 as a valid resting
3 position for the cursor (steps 1110 and 1130 of Figure 11).

4 At this point, the Determine Ideal Control Program of Figure 11 examines
5 the Time Entry control 753 of Figure 7B. In this case, the preferred embodiment
6 determines that the associated absolute distance 793 for the Time Entry control
7 753 is equal to 500 units (step 1140). Thus, the absolute distance to the Time
8 Entry control is greater than the skip distance value of 450. As a result, the
9 Determine Ideal Control Program does not consider the Time Entry control 753 as
10 a valid resting position for the cursor.

11 Additionally, the Determine Ideal Control Program of Figure 11 examines
12 the OK control 752 and the Cancel control 751 of Figure 7B (steps 1110-1170). In
13 contrast to the previously examined controls, these two controls are neither
14 obscured or out of respective distance or angle ranges. Therefore, the preferred
15 embodiment Position calculates the weighted distance for these two remaining
16 control candidates (step 1150). More specifically, assuming that the Control
17 Angle 781 for the Cancel control 751 is equal to 20° and that the absolute distance
18 connecting the point of release 780 with the center of the Cancel control 751 is
19 300 units, the associated Weighted Distance equals 360. [Weighted distance =
20 Absolute Distance + Control Angle/Weight Angle x Absolute Distance = 300 +
21 (20*300)/100 = 360] (step 1150). Since the Cancel control has an associated
22 weighted distance which is less than the value (65,535) held in the Current
23 Minimum Weighted Distance Parameter (step 1160), the preferred embodiment
24 sets the Current Minimum Weighted Distance Parameter to the Weighted Distance
25 associated with the Cancel control 751 (step 1170). Similarly, assuming that the

1 associated absolute distance for the OK control 752 of Figure 7B is 400 units and
2 the associated weight angle is 45°, the preferred embodiment calculates an
3 associated weighted distance of 580 (steps 1110-1150). In this case, however, the
4 calculated weighted distance is not less than the value held in the Current
5 Minimum Weighted Distance Parameter (step 1160). Therefore, the Determine
6 Ideal Control Program returns to the Position Cursor Program of Figure 10 with
7 the Cancel control 751 of Figure 7B identified as the control having the smallest
8 associated weighted distance. At this point, the Position Cursor Program positions
9 the cursor on the Cancel control 751 (step ¹⁰⁴⁰~~1140~~). In this manner, the preferred
10 embodiment of the present invention automatically places a cursor on an ideal
11 control. By determining an ideal control in this fashion, the preferred embodiment
12 allows the user to easily navigate between controls using only a crude positioning
13 device. For example, using either a four-arrow keypad or a joystick, the user can
14 effectively jump to a desired control by only momentarily activating the
15 directional control device in the direction of the desired control.

16 As explained above, the preferred embodiment of the present invention
17 determines the ideal control after the user has released the directional control.
18 However, those skilled in the art will appreciate that the preferred embodiment can
19 be easily modified to determine the ideal control while the user is moving the
20 cursor. In this alternative embodiment, the present invention periodically
21 calculates the weighted distances of the controls and designates the control having
22 the smallest weighted distance as a current ideal control. The alternative
23 embodiment then positions the cursor on the current ideal control when the user
24 releases the directional control device.

00527.03098
B225060

1 Additional advantages of, and modifications to, the preferred embodiment
2 of the present invention will readily appear to those skilled in the art. Therefore,
3 the broader aspects of the present invention are not limited to the specified details
4 of the described preferred embodiment. Accordingly, various modifications may
5 be made to the preferred embodiment without departing from the spirit and scope
6 of the present invention. As such, the present invention is only limited by the
7 appended claims and their equivalents.

8 Although the invention has been described in language specific to structural
9 features and/or methodological steps, it is to be understood that the invention
10 defined in the appended claims is not necessarily limited to the specific features or
11 steps described. Rather, the specific features and steps are disclosed as preferred
12 forms of implementing the claimed invention.
13
14
15
16
17
18
19
20
21
22
23
24
25